JUNITMZ at SemEval-2016 Task 1: Identifying Semantic Similarity Using Levenshtein Ratio

Sandip Sarkar
Computer Science and Engineering
Jadavpur University, Kolkata
sandipsarkar.ju@gmail.com

Dipankar Das
Computer Science and Engineering
Jadavpur University, Kolkata
dipankar.dipnil2005@gmail.com

Partha Pakray
Computer Science and Engineering
NIT Mizoram, Mizoram
parthapakray@gmail.com

Alexander Gelbukh
Center for Computing Research
Instituto Politcnico Nacional, Mexico
gelbukh@gelbukh.com

Abstract

In this paper we describe the JUNITMZ [1] system that was developed for participation in SemEval 2016 Task 1: Semantic Textual Similarity. Methods for measuring the textual similarity are useful to a broad range of applications including: text mining, information retrieval, dialogue systems, machine translation and text summarization. However, many systems developed specifically for STS are complex, making them hard to incorporate as a module within a larger applied system.

In this paper, we present an STS system based on three simple and robust similarity features that can be easily incorporated into more complex applied systems. The shared task results show that on most of the shared tasks evaluation sets, these signals achieve a strong (>0.70) level of correlation with human judgements. Our system's three features are: unigram overlap count, length normalized edit distance and the score computed by the METEOR machine translation metric. Features are combined to produces a similarity prediction using both a feedforward and recurrent neural network.

1 Introduction

Semantic similarity plays important role in many natural language processing (NLP) applications. The semantic textual similarity (STS) shared task has been held annually since 2012 in order to assess different approaches to computing textual similarity across a variety of different domains.

Research systems developed specifically for the STS task have resulted in a progression of systems that achieve increasing levels of performance but that are often also increasingly more complex. Complex approaches may be difficult if not impossible to incorporate as a component of larger applied NLP systems.

The system described in this paper explores an alternative approach based on three simple and robust textual similarity features. Our features are simple enough that they can be easily incorporated into larger applied systems that could benefit from textual similarity scores. The first feature simply counts the number of words common to the pair of sentences being assessed. The second provides the length normalized edit distance to transform one sentence into another. The final feature scores the two sentences using the METEOR machine translation metric. The latter allows the reuse of the linguistic analysis modules developed within the machine translation community to assess translation quality. METEOR's implementation of these modules is lightweight and efficient, making it not overly cumbersome to incorporate features based on METEOR into larger applied systems.

The remainder of this paper is structured as follows. Section 2 provides an overview of our system architecture and Section 3 describes our feature set. Section 4 reviews the neural network models we use to predict the STS scores. Section 5 describes the evaluation data followed by our results on the evaluation data in Section 6.

[1] This work was supported by the Project No. YSS/2015/000988 of National Institute of Technology Mizoram.

Proceedings of SemEval-2016, pages 702–705,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

2 System Framework

As shown in Figure 1, our system performs on a neural network based regression over our three textual similarity features. As described in the next section, the three similarity features we use are: unigram overlap count, editdistance and the METEOR score from the machine translation evaluation metric research community. The three features are combined using a neural network in order to predict a pair's final STS score.

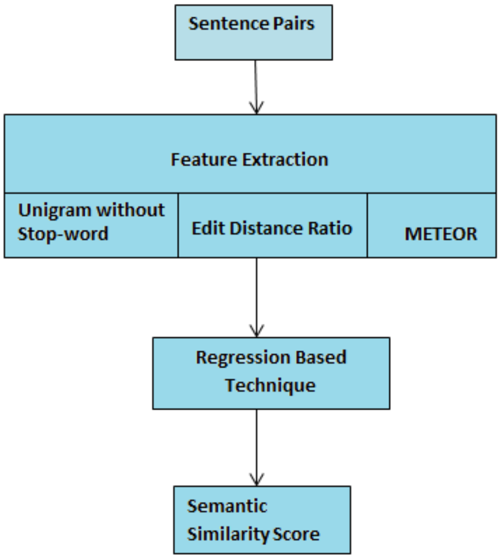
Sentence Pairs
Feature Extraction
Unigram without Stop-word
Edit Distance Ratio
METEOR
Regression Based Technique
Semantic Similarity Score

Figure 1: JUNITMZ STS System Architecture

3 Features

3.1 Unigram matching without stop-word

The unigram overlap count feature indicates the number of non stop-words that appear in both sentence pairs. [2] Table 1 illustrates the operation of this feature on an STS sentence pair.

The words ”to” and ”on” are present in both sentences, but we excluded them as stopwords for the purposes of the unigram overlap count.

Sentence Pair	Score
TSA drops effort to **allow small knives** on **planes**.	6
TSA drops plan to **allow small knives** on **planes**.	

Table 1: Unigram matching, ignoring stopwords

3.2 Edit Distance Ratio

We compute the minimum number of edit operations involving the insertion, deletion or substitution of individual characters that are required to transform one sentence into another. Commonly known as the Levenshtein distance (Levenshtein, 1966), this string similarity metric captures both similarity in the overall structure of the two sentences being compared as well as some similarity between different word forms (e.g., ”California” vs. ”Californian”).

As shown in equation (1) , we normalize the raw edit distance by the length of the two sentences. The score is then inverted such that a perfect match will have a score of 1.0, and completely dissimilar strings will be assigned a value of 0.0.

$$\mathrm{EditRatio}(a, b) = 1 - \frac{\mathrm{EditDistance}(a, b)}{|a| + |b|} \qquad (1)$$

An example of the edit distance ratio feature is given in Table 2.

Sentence Pair	Levenshtein Distance	Edit Distance Ratio
TSA drops effort to allow small knives on planes.	6	.8958
TSA drops plan to allow small knives on planes.		

Table 2: Edit Distance Ratio

3.3 Meteor

METEOR is a well known evaluation metric from the machine translation community (Denkowski and Lavie, 2014). The method incorporates linguistic analysis modules but in a manner that is lightweight, efficient and robust to the noisy data generated by machine translation systems. The method operates by first computing an alignment between the individual words in a sentence pair. In additional to matching identical words with each other, METEOR also supports matching words based on synonymy relationships in WordNet, entries in a paraphrase database or by word stem. The metric

[2] We obtain our stop word list from http://www.nltk.org/book/ch02.html

then computes a weighted F score based on the unigram alignments that is then scaled by a word scrambling penalty. The synonym matching is computed using WordNet. We use the METEOR 1.5 system for our STS Task.

4 Neural Network Framework

We predict STS scores based on three similarity features as described above using Matlab toolkit containing modules for three different neural networks. The neural networks have been used with respect to each of the corresponding runs submitted by our team to the shared task. The inputs of those network were the feature set along with the gold standard similarity scores extracted from the training data whereas the outputs produce the semantic scores for the test dataset. In **Run1**, we use two-layer feedforward network with 10 neurons in the hidden layer and trained using the Levenberg-Marquardt algorithm. [3] **Run2** uses the same network but trained using Resilient Backpropagation algorithm (Riedmiller and Braun, 1992). [4] In case of **Run 3**, we use the framework of Layer Recurrent Network which can be seen as a generalization of simple recurrent networks (Elman, 1990). [5]. The inputs of this recurrent neural network were similar like the other 2 neural network with default parameter.

5 Dataset

The 2016 STS shared task includes sentence pairs from a number of different data sources organized into five evaluation sets: News Headlines, Plagiarism, Postediting, Q&A Answer-Answer and Q&A Question-Question. The sentence pairs are assigned similarity scores by multiple crowdsourced annotators on a scale ranging from 0 to 5 with the scores having the following interpretations: (5) complete equivalence, (4) equivalent but differing in minor details, (3) roughly equivalent but differing in important details (2) not equivalent but sharing some details (1) not equivalent but on the same topic (0) completely dissimilar. The individual crowdsourced scores are aggregated to assign a final gold standard similarity score to each pair.

Sentence Pairs	Score
Two green and white trains sitting on the tracks.	4.4
Two green and white trains on tracks.	
A cat standing on tree branches.	3.6
A black and white cat is high up on tree branches.	
A woman riding a brown horse.	3.8
A young girl riding a brown horse.	

Table 3: Examples of sentence pairs with their gold scores (on a 5-point rating scale)

Type	Sentence Pair
answer-answer	1572
headlines	1498
plagiarism	1271
postediting	3287
question-question	1555

Table 4: Statistics of STS-2016 Test Data

Table 3 provides example sentence pairs with their corresponding gold standard similarity scores. Systems are assessed on each data set based on the Pearson correlation between the scores they produce and the gold standard. The detailed statistics of the STS-2016 Test datasets are given in Table 4. For the training process we used all gold standard training and test data of year 2012 to 2015 resulting in 12500 sentence pairs.

6 Result

For our training dataset we use trail, training and test data from previous STS competitions. As shown in Table 5, we used different subsets of the data from prior STS evaluations to train different models for the 2016 evaluation sets.

Table 7 illustrates the performance of our three system submission on each of the STS 2016 evaluation sets as assessed by their correlation with the gold standard similarity scores. Overall performance is reported as the weighted mean correlation across all five data sets. The best overall correlation we obtain is 0.62708, which is achieved by run1, the LevenbergMarquardt trained feedforward network. For comparison, the best and mean scores achieved by all systems submitted to the 2016 STS shared task

[3] we have used Matlab for regression http://nl.mathworks.com/help/nnet/ref/feedforwardnet.html

[4] http://nl.mathworks.com/help/nnet/ref/trainrp.html

[5] http://nl.mathworks.com/help/nnet/ug/design-layer-recurrent-neural-networks.html

Test Dataset	Training Dataset	Count
answer-answer	MSRpar, MSRvid, OnWN, image	5350
headlines	MSRpar, MSRvid, SMTnews, headline	4899
plagiarism	MSRpar, MSRvid, OnWN, tweet-news	5250
postediting	MSRpar, OnWN, SMTnews	4193
question-question	MSRpar,OnWN, SMTeuroparl	4093

Table 5: Training data used for the STS-2016 datasets

Dataset	Best	Median
ALL	0.77807	0.68923
answer-answer	0.69235	0.48018
headlines	0.82749	0.76439
plagiarism	0.84138	0.78949
postediting	0.86690	0.81241
question-question	0.74705	0.57140

Table 6: Top and median scores of SemEval-2016

Dataset	Run1	Run2	Run3
ALL	**0.62708**	0.58109	0.59493
answer-answer	**0.48023**	0.40859	0.44218
headlines	**0.70749**	0.66524	0.66120
plagiarism	0.72075	**0.76752**	0.73708
postediting	**0.77196**	0.66522	0.69279
question-question	**0.43751**	0.38711	0.43092

Table 7: System performance on SemEval STS-2016 data.

are provided by Table 6.

While our models are less accurate in their predictions than other systems, we note that our submission is based on simple and robust features that allow it to be more easily integrated into complex downstream applications. With our feature set, we still achieve a strong (>0.70) correlation with human judgements on 3 of the 5 shared task evaluation sets. However, our system struggles on both of the Q&A data sets, questionquestion and answeranswer, suggesting additional signals may be necessary in order to correctly handle pairs from this domain.

7 Conclusion and Future Work

We have presented an STS system based on three simple robust features. The results of the shared task evaluation show that our feature set is able to achieve a strong (>0.70) correlation on 3 of the 5 shared task evaluation sets. The simplicity of our feature set should make it easier to incorporate into downstream applications. We do note that, similar to submissions from other teams, our systems struggle on the two question answering datasets. We are optimistic that it is possible to also obtain strong correlations on this dataset without resorting to overly complex systems.

In future we plan to investigate using features directly based on resources such as WordNet as well as attempt to generalize our system to the crosslingual formulation of the STS task.

Acknowledgments

I am very thankful to JUNLP team member for their help in my SEMEVAL task. I am also thankful to the organizer for their support. This work was supported by the Project No. $YSS/2015/000988$ of National Institute of Technology Mizoram.

References

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Jeffrey L. Elman. 1990. Finding structure in time. *COGNITIVE SCIENCE*, 14(2):179–211.

V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, February.

M. Riedmiller and H. Braun. 1992. RPROP: A fast adaptive learning algorithm. In E. Gelenbe, editor, *International Symposium on Computer and Information Science VII*, pages 279 – 286, Antalya, Turkey.